Software Design Specification

Alejandro Thompson, Alberto Escalante , Isaac Reveles

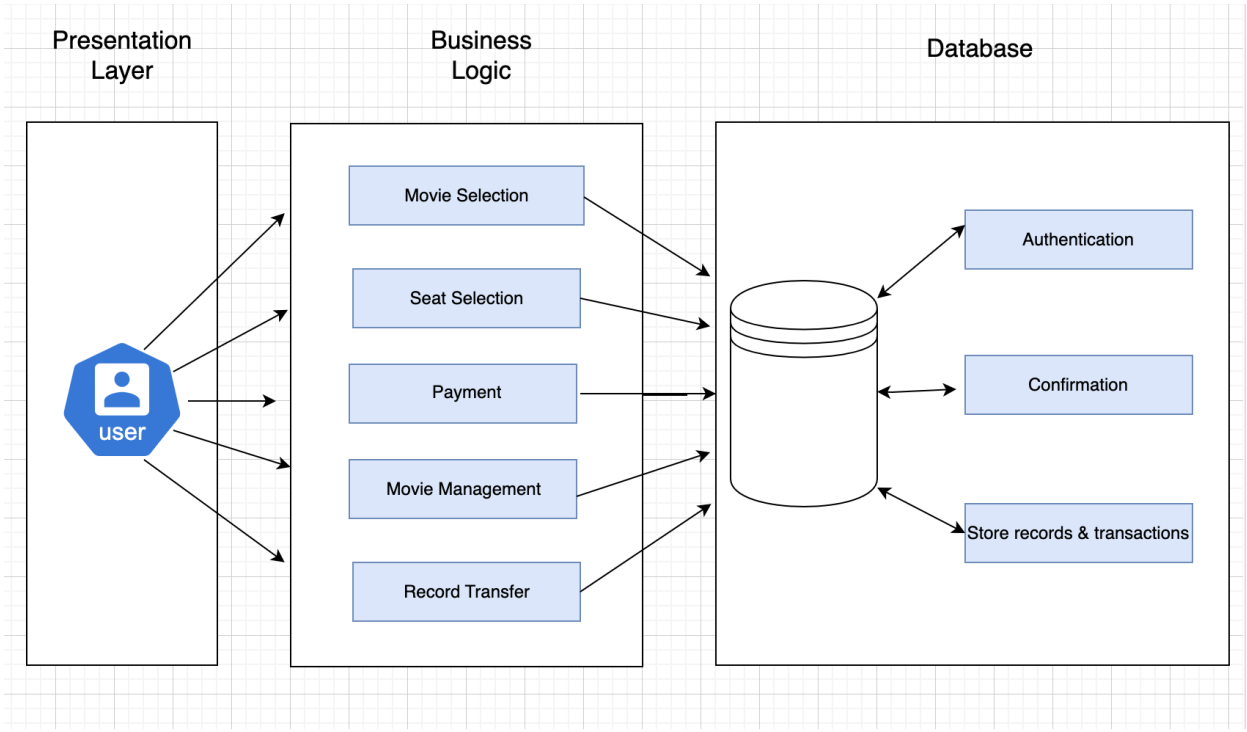
System Description

Brief overview of the system:
The SD Theater Ticketing System is an online system designed using an architectural diagram and a UML diagram. This Software Design Specification is intended to be a comprehensive guide that will showcase the detail that goes into the architecture, functionality and overall the structure of the system. This documentation is for the developers to see the correlation between each component of the ticketing system.

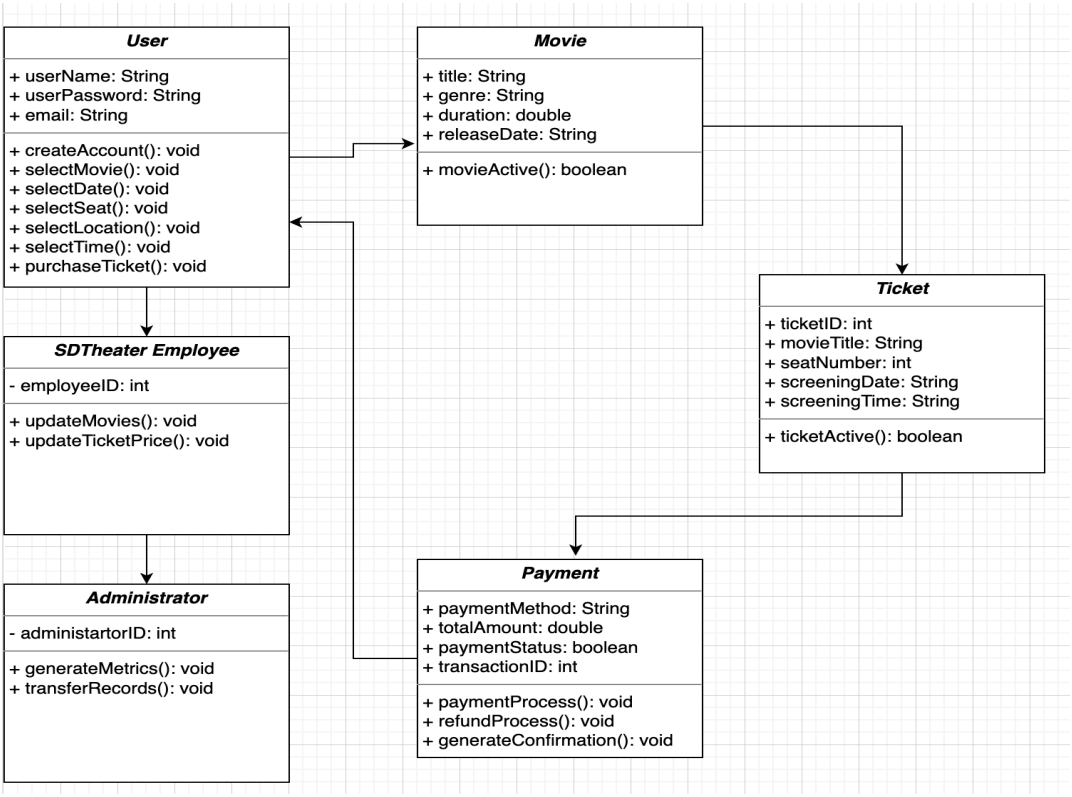
Software Architecture Overview

For the architecture of the system, it is based on the 3-layered data access approach; which are in this case Presentation Layer, Business Logic, and Database. The architecture below showcases the correlation between each layer.

Architecture Diagram:



UML Class Diagram:



Description of classes:

User: Represents users of the system, including customers and administrators. This class may have attributes such as ***Username, Password, Email, Role, etc.***

Admin: Represents an administrator of the system responsible for managing performances, venues, tickets, etc. This class may have attributes such as ***Username, Password, Permissions, etc.***

Movie: Represents the showing of the movie. This class may have attributes such as ***Title, Date, Time, Venue, Description of movie, etc.***

Employee: Represents the employees of the SD Theatre. This class may have attributes such as ***Employee ID, etc.***

Ticket: Represents a ticket purchased by a customer for a specific movie. This class may have attributes such as ***Ticket ID, Movie ID, Price, etc.***

Payment: Represents a payment transaction made by a customer to purchase tickets. This class may have attributes such as ***Payment ID, Amount, Payment Method, Payment Status, etc.***

Description of Attributes:

User:

1.) **userName**: User chooses their preferred name for their SD Theater account. Represented as a String.
2.) **userPassword**: User enters their password for verification. Represented as a String.
3.) **email**: User's email address where they will be able to receive notifications and confirmations in regards to their account. Represented as a String.

SDTheaterEmployee:

1.) **employeeID**: SD Theater employees have the same user controls, the only additional information presented is their unique employee ID which gives them staff access to the system. Represented as an int that is private.

Administrator:

1.) **administratorID**: Administrators have the same access as customers and theater employees, the only new information needed is their administrator ID in order to have administrator access. Represented as an int that is private.

Movie:

1.) **title**: The name of the movie. Represented as a String.
2.) **genre**: The classification of the movie. Represented as a String.
3.) **duration**: The time span of the movie. Represented as a double.
4.) **releaseDate**: The date the movie will be released to the public in theaters. Represented as a String.

Ticket:

1.) **ticketID**: The unique ID for every ticket. Represented as an int.
2.) **movieTitle**: The name of the movie associated with the ticket. Represented as a String.
3.) **seatNumber**: Unique number of the seat associated with the ticket. Represented as an int.
4.) **screeningDate**: The date of when the movie will be played. Represented as a String.
5.) **screeningTime**: The time at which the movie will be played. Represented as a String.

Payment:

1.) **paymentMethod**: The source of payment the customer will use to pay. Represented as a String.
2.) **totalAmount**: The total amount that is due at checkout. Represented as a double.
3.) **paymentStatus**: Determines if the payment was accepted. Represented as a boolean.
4.) **transactionID**: The unique ID of every transaction made. Represented as an int.

Description of Operations:

User:

1.) **createAccount(): void**: This operation allows the user to create an SD Theater account by entering their username, password, and email address.
2.) **selectMovie(): void**: Enables the user to select any movie.
3.) **selectDate(): void**: Enables the user to select a date associated with the movie.
4.) **selectSeat(): void**: Enables the user to select any seat(s) that is/are available.
5.) **selectLocation(): void**: Enables the user to select any of the SD Theater locations in San Diego.
6.) **selectTime(): void**: Enables the user to select from a list of time slots that are available.
7.) **purchaseTicket(): void**: Allows the user to purchase number of tickets; depending on how many children, adults, or seniors are attending, lastly is taken to checkout.

SDTheaterEmployee:

1.) **updateMovies(): void**: Enables access for SD Theater employees to update the movie listing.
2.) **updateTicketPrice(): void**: SD Theater employees are able to change the price of tickets.

Administrator:

1.) **generateMetrics(): void**: Creates metrics of movies and how they have performed within the theater and system.
2.) **transferRecords(): void**: Allows access for security hub to be accessed and for business transactions and records to be transferred into the database.

Movie:

1.) **movieActive(): boolean**: Gets the status of any movie. Returns true if movie is still playing in theaters. Returns false if movie is not in theaters anymore.

Ticket:

1.) **ticketActive(): boolean**: Returns true if user ticket is currently active within the system.

Payment:

1.) **paymentProcess():void**: Runs payment process operation.
2.) **refundProcess():void**: If user requires refund for tickets, this operation runs the refund process.
3.) **generateConfirmation():void**: Confirms payment of tickets.

Development plan and timeline

To make sure the development of the SDTheatre software goes smoothly, we will be following a structured timeline as follows:

• **Weeks 1-2:** Project Planning, Location Verifications and Resource Requirements

• **Weeks 3-4:** Project Design and Beginning of Software Development

• **Weeks 5-6:** User and Employee Registration, Account Info and Login System Integration

• **Weeks 7-8:** Implementation of Payment System and PearPay Integration

• **Weeks 9-10:** Location, Theater and Seat Implementation

• **Weeks 11-12:** Ticket System Implementation and IMDbRT Integration

• **Weeks 13-14:** Security Implementation and Alpha Testing

• **Weeks 15-16:** Bug Fixes and Final Testing

• **Week 17:** Documentation and Beta Testing

• **Week 18:** Release Software

Team member responsibilities:

The following tasks will be assigned to team members as follows:

• **Project Manager:** Oversees that all objectives in the timeline are completed within parameters and that team members are completing their tasks efficiently.

• **Business Analyst:** Gather information and scope out the best locations and theaters for SDTheater and whether it can be verified there. Also verifies if the product is ready for commercial use.

• **Engineering Manager:** In charge with working with other team members to ensure a safe and a working product. Responsible of analyzing any challenges in hardware and figuring out fixes for them.

• **Software Architect:** In charge of conducting the internal arrangement of the software and work with the Software Developers to perfect the product with optimal technical solutions.

• **Software Developers:** Responsible of implementing the internal arrangement and

systems in the software by working with the Software Architect and in parallel with UI Designers and QA Engineers.

• **UI Designers:** In charge of the Project Design and UI system of the software. Makes sure the user experience is optimal, easy and convenient to use. Also work in parallel with Engineer and Software Developers.

• **QA Engineer:** In charge of quality assurance during testing, makes sure to point out any flaws in the system and automate the testing process for software quality.

• **Testers:** Required to test the software in the Alpha and Beta phase to see if it is ready for commercial use. Provides feedback before the release of the software.